

Maximize the Benefits of Oracle SOA Suite 11g with Oracle Service Bus

An Oracle White Paper
July 2010

Maximize the Benefits of Oracle SOA Suite 11g with Oracle Service Bus

Table of Contents

Introduction.....	3
Challenges Addressed by SOA.....	3
SOA Concepts and Architectural Model	4
What is a Service?	4
Providers, Consumers and Service Contracts.....	4
SOA Architectural Model.....	5
Business Service Layers.....	6
SOA Infrastructure.....	6
Oracle SOA Suite 11g.....	7
Components of the Oracle SOA Platform.....	7
Addressing Key Customer Needs	7
Role of an Enterprise Service Bus Within SOA.....	9
Key values provided by a Service Bus	10
Capabilities of Oracle Service Bus	11
Differentiating between Oracle Service Bus and Mediator.....	13
Modeling Business Process Orchestration.....	14
Aligning Architecture and Project Teams.....	15
Summary	16

Maximize the Benefits of Oracle SOA Suite 11g with Oracle Service Bus

INTRODUCTION

Oracle's SOA Suite 11g provides a comprehensive set of tools for the development of an enterprise wide IT platform supporting core business processes. A service oriented approach to enterprise architecture promises agility, manageability, visibility, risk reduction and lower total cost of ownership. This paper provides an outline of how the various components within Oracle SOA Suite 11g and Oracle Service Bus in particular can be leveraged to realize the above benefits.

CHALLENGES ADDRESSED BY SOA

The current economic conditions and rapid changes in business environment require businesses to adapt quickly while reducing costs in the longer term. From an IT perspective, the above requirements translate to increasing operational efficiency while simultaneously increasing agility and visibility. Increase in operational efficiency is being driven through further process automation and system consolidation. However, integrating multiple systems can lead to complex and brittle architectures that can be very expensive to change and maintain. Some of the key goals that IT organizations need to address in this context are:

A SOA platform is not just about web services based integration; it is about providing an architecture and technology platform that solves fundamental business challenges faced within complex enterprise-wide IT projects.

- Reduce time to market for new functionality
- Reduce integration cost and complexity
- Manage business and technology change
- Streamline business process exception handling
- Provide end-to-end solution monitoring with root cause analysis
- Enable faster reaction to business events through increased visibility
- Ensure high availability and scalability of the digitized platform

SOA CONCEPTS AND ARCHITECTURAL MODEL

To understand how a SOA technology platform addresses the above challenges, it is important to begin with the basic concepts of a service oriented approach.

Service Oriented Architecture (SOA) is a strategy for constructing business-focused, software systems from loosely coupled, interoperable building blocks (Services) that can be combined and reused quickly, within and between enterprises, to meet business needs.

What is a Service?

The basic concept within SOA is that of a **Service**. A service in the context of SOA can be described as a self-sufficient functional endpoint that has a well defined functionality and service level agreement, can be monitored and managed, and can help enforce policy compliance.

From a technical standpoint a Service is comprised of three parts:

- the implementation (deployed code and configuration of infrastructure),
- the interface (means by which the Service is invoked), and
- the contract (a description of what the Service provides and its constraints)

Providers, Consumers and Service Contracts

A Service is implemented by a Service Provider and is exposed to one or more Service Consumers through an interface. Consumers of a service may be internal applications such as business processes and web portals or external to the organization. The Service Contract between the provider and consumer of a service not only includes the programmatically enforceable interface in terms of document structure, security policies etc. but can also include non-functional aspects such as availability and latency specifications.

Business and technology changes drive new implementations of an existing service. However, it is important to maintain the existing Service Contract to ensure business continuity while incorporating the new implementation.

In the context of SOA, the interface of a service which is a part of the Service Contract typically has a longer lifespan than the implementation of the same service. Thus, there is a need to create an architecture that enables the separation of the lifecycle of a service interface and contract from the implementation.

This above principle can be applied at different levels of enterprise architecture to contain and minimize the impact of local changes to larger integrated systems.

In the context of Enterprise architecture, Service-orientation, and Service-oriented architecture, the term service refers to a set of related software functionality, together with the policies that should control their usage. ... A Web service provides one way of implementing the automated aspects of a given business or technical service."

- Wikipedia

SOA Architectural Model

The intent of SOA is to provide common reusable Services that can be leveraged by a variety of consumers. These reusable services typically originate from functionality and data that already exist within the enterprise; i.e. services created by service-enabling legacy assets. As new projects are implemented, standalone Services may also emerge as autonomous entities that do not have dependencies on legacy systems.

An architectural model for SOA can be divided into two high level categories:

- Business Service Layers
- SOA Infrastructure

This categorization provides a separation of IT concerns and business logic concerns and that is critical to the success of a SOA implementation. Separating concerns into these layers allows greater agility and flexibility in the architecture.

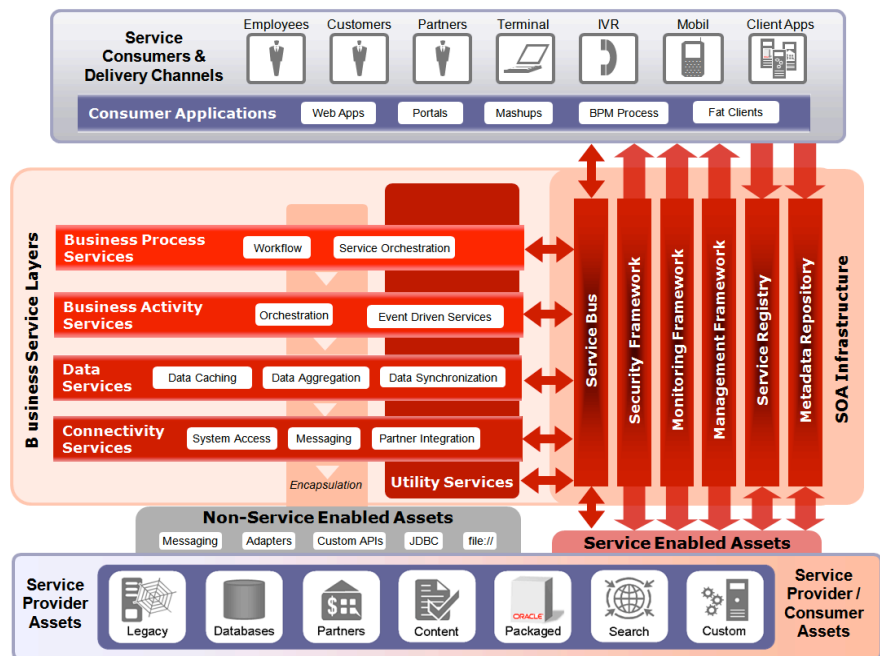


Figure 1 - An Architectural Perspective of a SOA Platform

Business Service Layers

The Business Service Layers host services that implement application logic from discrete business activities to complete business processes. The following functional capabilities are addressed by this category:

- Business process orchestration
- Business exception management
- Human interaction with business processes
- Business rules modeling
- Structured conversational interactions with other systems and trading partners

In summary, this layer deals with concerns associated with process automation and business logic development. It does not address IT concerns such as abstracting the physical location of a service endpoint or ensuring end-to-end resilience to system failures. This layer also addresses business level visibility issues by integrating with business events for event driven SOA.

SOA Infrastructure

The SOA Infrastructure provides capabilities to publish, discover, share, monitor and manage services. Services utilize the connection management and recovery functions provided by the SOA infrastructure. The infrastructure must be scalable to support the requirements on involved parties regarding performance, response time, availability, network connections and capacity. The SOA Infrastructure decouples service providers from consumers. Decoupling enables location transparency and ability to introduce new versions of a service without impacting all clients of the service.

In summary, this category addresses IT concerns around design time and runtime visibility, governance of assets and policy enforcement. This category typically does host application logic but can enable higher granularity services through simple service aggregation.

ORACLE SOA SUITE 11G

Oracle SOA Suite 11g provides a comprehensive integration suite of products that together provide the necessary capabilities to address the strategic IT goals identified above.

Components of the Oracle SOA Platform

The key components within Oracle SOA platform include:

- Oracle SOA Composite Platform
 - BPEL Process Manager (BPEL PM)
 - Human Workflow (HWF)
 - Business Rules
 - Mediator
- Oracle Service Bus (OSB)
- Oracle Business Activity Monitoring (BAM)
- Oracle Complex Event Processing (CEP)
- Oracle Enterprise Manager (EM)
- Oracle Web Services Manager (OWSM)
- Oracle Enterprise Repository and Registry (OER / OSR)
- Oracle Data Integrator (ODI)

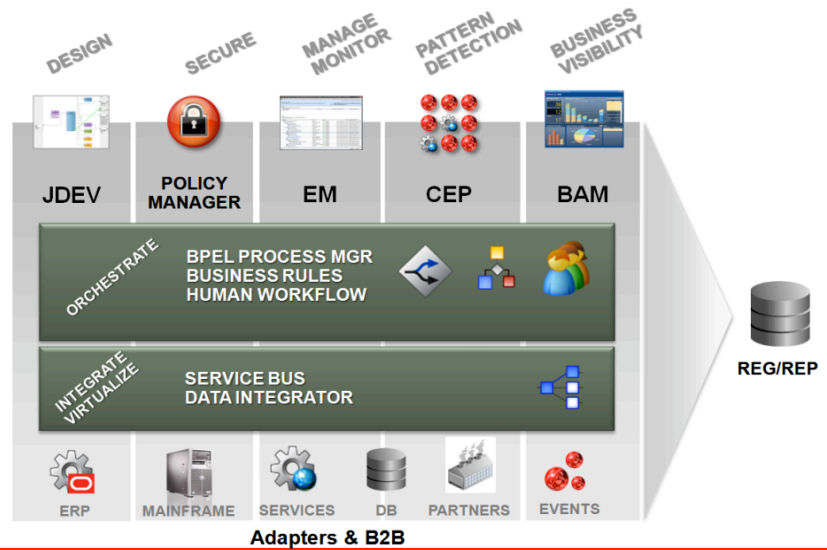


Figure 2 - Components of SOA Suite 11g

Addressing Key Customer Needs

Presented below is a mapping of the identified customer needs to the capabilities of various components within SOA Suite 11g. These capabilities may be obtained through the deployment of a single component or accrued through the utilization of multiple components leveraging the common infrastructure.

Customer Need	Capabilities of Oracle SOA Suite 11g
Reduce time to market for new functionality	<ul style="list-style-type: none"> • <i>Rapid configuration driven development model</i> • <i>Declarative business process modeling using BPEL and business rules</i> • <i>SCA based assembly of SOA components into composite applications</i> • <i>Ability to easily leverage existing J2EE / Java assets</i>
Reduce integration cost and complexity	<ul style="list-style-type: none"> • <i>JCA based declarative connectivity to extensive set of technologies, applications and legacy systems</i> • <i>Integrated support for B2B agreements and document templates</i> • <i>Tooling support for key integration and message exchange patterns</i> • <i>Ability to share services and reduce point-to-point connections</i>
Manage business and technology change	<ul style="list-style-type: none"> • <i>Service Bus for location transparency and service mediation (shape, transport protocol, interaction pattern, security paradigm)</i> • <i>Decouple frequently changing rules from core process logic</i> • <i>Shared policy based security enforcement</i> • <i>In-built versioning support</i>
Streamline business process exception handling	<ul style="list-style-type: none"> • <i>Seamless integration between automated processes and human workflows for process exception management</i> • <i>Policy based error handling</i> • <i>Error hospital capabilities</i>
Provide end-to-end solution monitoring with root cause analysis	<ul style="list-style-type: none"> • <i>Cluster-wide management and monitoring console</i> • <i>Integrated monitoring and SLA based alerting</i> • <i>End-to-end instance tracking with state and error information</i> • <i>Ability to drill-down into audit trails of individual components</i>
Enable faster reaction to business events through increased visibility	<ul style="list-style-type: none"> • <i>Real-time business activity monitoring (BAM) dashboards</i> • <i>Event driven alerting capabilities</i> • <i>Complex Event Processing - Real time pattern matching on high throughput event streams</i>
Ensure high availability and scalability of the digitized platform.	<ul style="list-style-type: none"> • <i>Scalable & high performance service bus based SOA infrastructure</i> • <i>Traffic shaping – throttling, prioritization, parallel execution</i> • <i>Scalable data access using Coherence distributed caching capabilities</i>

While Oracle SOA Suite 11g provides a rich set of capabilities, it is important to align the implementation strategy with the architectural guidelines described previously in order to realize some of the described benefits. A good starting point is to understand the categorization of the SOA Suite 11g components into architectural categories identified above. The set of components may be classified as:

Business Service Layers	<ul style="list-style-type: none"> • Oracle SOA Composite Platform <ul style="list-style-type: none"> - BPEL Process Manager (BPEL PM) - Human Workflow (HWF) - Business Rules - Mediator • Oracle Business Activity Monitoring (BAM) • Oracle Data Integrator (ODI)
SOA Infrastructure	<ul style="list-style-type: none"> • Oracle Service Bus (OSB) • Oracle Complex Event Processing (CEP) • Oracle Enterprise Manager (EM) • Oracle Web Services Manager (OWSM) • Oracle Enterprise Repository and Registry (OER / OSR)

Note that the above is a rough categorization of products. Certain products can be used in both categories but with different end-goals. For example, OWSM can be used to enforce a security policy on a shared service exposed through the SOA Infrastructure. OWSM may also be used within the service layers to ensure single sign-on.

The above categorization also does not reflect the deployment topology of the SOA infrastructure. A certain capability such as monitoring may span multiple servers and be exposed through Enterprise Manager.

ROLE OF AN ENTERPRISE SERVICE BUS WITHIN SOA

An Enterprise Service Bus (ESB) is a critical component within the SOA Infrastructure category that integrates with all other components within the architecture. An ESB forms the basis for a shared services platform and is the stepping stone to effective governance.

Key values provided by a Service Bus

Within the context of SOA, an ESB provides the following high-level values.

- **Loose Coupling of Services**

Loose coupling is achieved through abstracting and resolving the differences between two or more systems in order to facilitate a seamless integration. An ESB provides help mediate the differences along the following lines:

- Transport Protocol
- Interaction Pattern (message exchange pattern)
- Shape (Interface / Schema)
- Security Paradigm

- **Service Enablement**

Service enable refers to leveraging existing assets to create an end-point that meets the definition of a service described previously in this paper. Service enablement through an ESB provides the following values:

- Location transparency
- Policy attachment and enforcement
- Service granularity management through service aggregation

- **Change Management**

Change management refers to ability the change various attributes of a service without impacting existing clients.

- Add new consumers to an existing service
- Change service implementation
- Change policies associated with a service
- Add new versions of the service
- Add additional providers of a service

Forrester defines an Enterprise Service Bus (ESB) as “an intermediary that provides core functions to make a set of reusable services widely available, plus extended functions that simplify the use of the ESB in a real-world IT environment.”

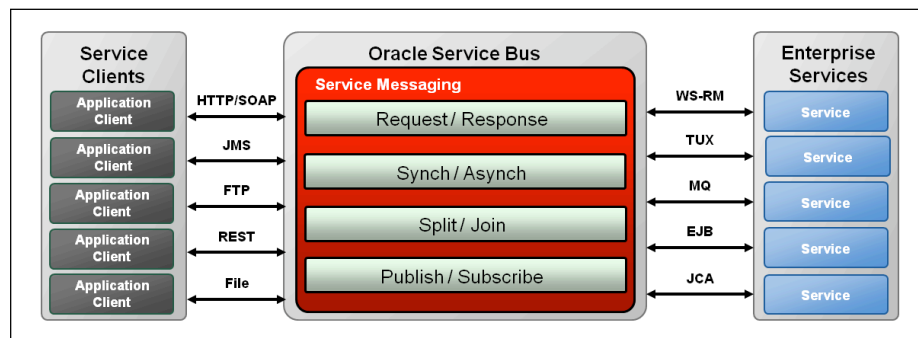


Figure 3 – Service Mediation using an ESB

Capabilities of Oracle Service Bus

Oracle Service Bus is the strategic enterprise-level ESB platform within the Oracle SOA Suite. It delivers the key values of an ESB through the following capabilities:

- Routing (Static / Dynamic)
- Shape (schema) transformation
- Protocol (transport) switching
- Monitoring and SLA based alerting sub-system that integrates with Enterprise Manager
- Service lifecycle management (versioning, registry integration etc.) automatically synchronized with Oracle Enterprise Repository and Service Registry.
- Security policy enforcement and mediation through the support of multiple policy providers including OWSM
- Service Result Caching, through embedded access to Oracle Coherence, to eliminate latency time associated with repeated access to static back-end data with easy to use, single-click enablement.
- Auditing
- Service aggregation
- Traffic Shaping
 - Throttling
 - Prioritization
 - Parallel Processing
- End Point Management
 - Service pooling and load-balancing
 - System fault isolation and recovery

Forrester notes: "most SOA efforts rely heavily on one or more ESBs as their core software infrastructure"

Thus, Oracle Service Bus provides a powerful platform to address some of the important IT infrastructure needs within enterprise architecture.

Presented below are diagrams that explain the value of virtualizing a service using Oracle Service Bus.



Figure 4 - Basic Service Invocation

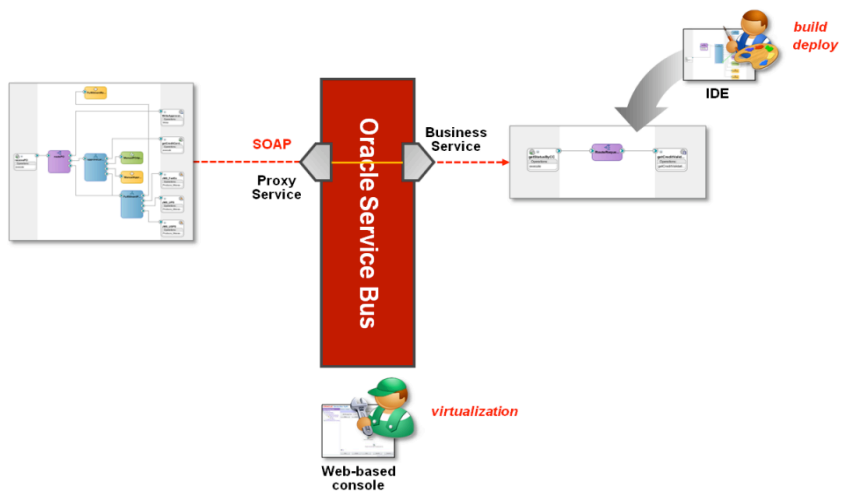


Figure 5 – Service Virtualization using Oracle Service Bus

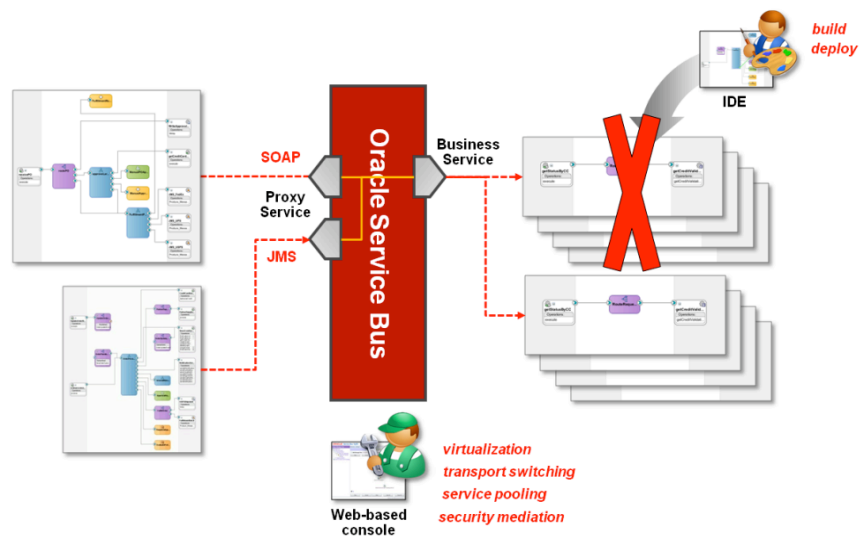


Figure 6 - Service virtualization using Oracle Service Bus with automated service pooling, transport switching, and mediation

Differentiating between Oracle Service Bus and Mediator

While Oracle Service Bus provides enterprise service re-use and management; the Mediator component provides certain localized mediation capabilities with the Business Service Layer. Thus, the lifecycle of a Mediator component is tightly coupled with that of the SOA composite application that provides the application logic. Mediator provides the following capabilities with the context of a single composite application:

- Connectivity abstraction from a business process
- Inline data transformation / mapping
- Message filtering

Oracle Service Bus enables effective de-coupling of systems and lifecycles within enterprise architecture. Mediator provides any abstraction that the Business Process needs within the context of a single composite. The key considerations for using Mediator include:

- The functionality is available within the context of a single SOA composite application. E.g. Mediator can be used to expose a BPEL process to multiple services defined on the same composite.
- Mediator does not focus on key capabilities required for the SOA Infrastructure category such as traffic shaping and end-point management.
- Mediator should not be used to share services at an enterprise-wide level. Since the lifecycle of a Mediator component is tied to the lifecycle of the underlying process logic, Mediator cannot effectively separate the lifecycle of the service contract from the service implementation.

An example of the above principle is determining where to implement the message transformation logic associated with an integration scenario. When integrating between multiple systems/domains, there are two distinct approaches to data modeling that may be determined by the Information Architecture:

1. Point-to-point Data Mapping: The business objects are exposed in domain (system) specific format. Transformation logic maps the data format between the native formats of the 2 systems.
2. Normalized Domain Model: The business objects are always exposed in a normalized data format i.e. the data is always transformed to a common domain model when exposed through a service. Transformations are used to convert to/from the normalized format from / to the system specific format.

The difference in the above approaches is that the transformation logic in first approach is use case specific while the transformation logic in the second approach can be shared by all use cases using the same data. Thus, transformations that fit the first pattern can be implemented in Mediator while transformations that fit the second pattern should be implemented in Oracle Service Bus.

Modeling Business Process Orchestration

Any application logic should be delivered through one of the Business Service Layers. In the case of Oracle SOA Suite 11g, such logic should typically be implemented using the BPEL Process Manager. Oracle Service Bus is specialized to address use cases associated with the SOA Infrastructure category. While it is possible to create simple transactional service aggregation scenarios within Oracle Service Bus; due care must be exercised when going down this path as Oracle Service Bus is a stateless engine that is optimized for use cases with short-lived single transaction semantics. Below are examples of use cases that require a full-fledged orchestration engine and should not be implemented on Oracle Service Bus:

- Service needs to maintain state
- Service requires complex transaction management
- Requires multiple transactions
- Compensation logic required on rollback
- Short or long-lived process
- Exception handling requires Human workflow
- Service needs to handle asynchronous callbacks reliably

Thus, Oracle Service Bus provides a clean separation between clients, business processes and back-end information systems. This virtualization layer can be used to inject various IT concerns such as traffic shaping, alerting, and fault isolation.

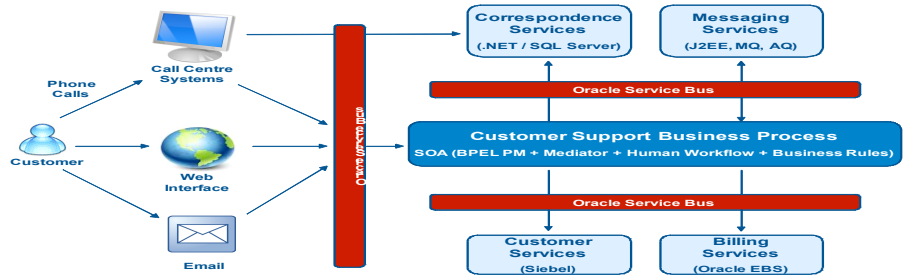


Figure 7 – Sample architecture using Oracle Service Bus with SOA Composite

ALIGNING ARCHITECTURE AND PROJECT TEAMS

A structured approach to a SOA project ensures alignment between people, technology and business processes. Figure-8 illustrates a typical project structure that enables the development of a robust SOA architecture and implementation methodology.

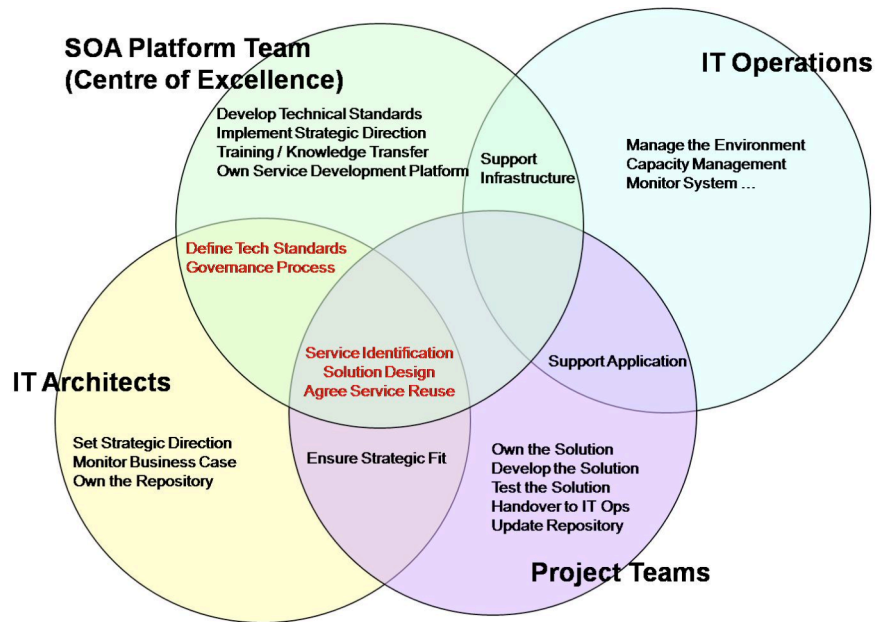


Figure 8 - Roles and Responsibilities in a SOA Project

A project team structure as described above is important to ensuring that key goals for the SOA project are being met. This is achieved by the separation of concerns between the line of business focused project teams and the IT focused architecture

and SOA platform teams. Adopting an architectural approach as described in this paper helps aligning the deliverable components to the individual teams. Implementation well defined governance processes is necessary to scale the described approach to a large number of services systems and teams.

SUMMARY

Oracle SOA Suite 11g provides a comprehensive set of tools that address the key challenges faced by IT organizations in today's market conditions.

While choosing the right technology platform is essential, it is important to note that adopting a service oriented approach to the IT architecture and development processes is as necessary as the underlying technology platform to obtain the benefits listed above. An Enterprise Service Bus is critical to creating a scalable SOA backplane that provides the foundation for virtualization, policy enforcement, monitoring, event visibility, service discovery and effective governance.



Maximize the Benefits of Oracle SOA Suite 11g with Oracle Service Bus

July 2010

Author: Samrat Ray

Contributors: Manas Deb, Deb Ayers, Andrea Morena

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

oracle.com

Copyright © 2010, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.